Extra Practice: R

GSND 5345Q, Fundamentals of Data Science

Due (not officially) Monday, January 26, 2026

R Basics

These exercises will give you some introductory experience with the R programming basics. Please complete the following:

1. What is the sum of the first 100 positive integers? The formula for the sum of integers 1 through n is $n(n + 1)/2$. Define $n = 100$ and then use R to compute the sum of 1 through 100 using the formula. What is the sum?

2. Now use the same formula to compute the sum of the integers from 1 through 1,000.

3. Look at the result of typing the following code into R:

```
n <- 1000
x <- seq(1, n)
sum(x)
```

Based on the result, what do you think the functions `seq` and `sum` do? You can use `help`.

    a. `sum` creates a list of numbers and `seq` adds them up.
    b. `seq` creates a list of numbers and `sum` adds them up.
    c. `seq` creates a random list and `sum` computes the sum of 1 through 1,000.
    d. `sum` always returns the same number.

4. In math and programming, we say that we evaluate a function when we replace the argument with a given number. So if we type `sqrt(4)`, we evaluate the `sqrt` function. In R, you can evaluate a function inside another function. The evaluations happen from the inside out. Use one line of code to compute the log, in base 10, of the square root of 100.

5. Which of the following will always return the numeric value stored in `x`? You can try out examples and use the help system if you want.

    a. `log(10^x)`
    b. `log10(x^10)`
    c. `log(exp(x))`
    d. `exp(log(x, base = 2))`

6. Load the US murders dataset.

```
library(dslabs)
```

```
## Warning: package 'dslabs' was built under R version 4.5.2
```

```
data(murders)
```

Use the function `str` to examine the structure of the `murders` object. Which of the following best describes the variables represented in this data frame?

a. The 51 states.
b. The murder rates for all 50 states and DC.
c. The state name, the abbreviation of the state name, the state's region, and the state's population and total number of murders for 2010.
d. `str` shows no relevant information.

7. What are the column names used by the data frame for these five variables?

8. Use the accessor `$` to extract the state abbreviations and assign them to the object `a`. What is the class of this object?

9. Now use the square brackets to extract the state abbreviations and assign them to the object `b`. Use the `identical` function to determine if `a` and `b` are the same.

10. We saw that the `region` column stores a factor. You can corroborate this by typing:

```
class(murders$region)
```

With one line of code, use the function `levels` and `length` to determine the number of regions defined by this dataset.

11. The function `table` takes a vector and returns the frequency of each element. You can quickly see how many states are in each region by applying this function. Use this function in one line of code to create a table of states per region.

12. Use the function `c` to create a vector with the average high temperatures in January for Beijing, Lagos, Paris, Rio de Janeiro, San Juan, and Toronto, which are 35, 88, 42, 84, 81, and 30 degrees Fahrenheit. Call the object `temp`.

13. Now create a vector with the city names and call the object `city`.

14. Use the `names` function and the objects defined in the previous exercises to associate the temperature data with its corresponding city.

15. Use the `[` and `:` operators to access the temperature of the first three cities on the list.

16. Use the `[` operator to access the temperature of Paris and San Juan.

17. Use the `:` operator to create a sequence of numbers $12, 13, 14, \ldots, 73$.

18. Create a vector containing all the positive odd numbers smaller than 100.

19. Create a vector of numbers that starts at 6, does not pass 55, and adds numbers in increments of 4/7: 6, 6 + 4/7, 6 + 8/7, and so on. How many numbers does the list have? Hint: use `seq` and `length`.

20. What is the class of the following object `a <- seq(1, 10, 0.5)`?

These exercises will give you some introductory experience with programming basics. Please complete the following:

21. What will this conditional expression return?

```
x <- c(1,2,-3,4)

if(all(x>0)){
  print("All Postives")
} else{
  print("Not all positives")
}
```

22. Which of the following expressions is always `FALSE` when at least one entry of a logical vector `x` is TRUE?

    a. `all(x)`
    b. `any(x)`
    c. `any(!x)`
    d. `all(!x)`

23. Create a function `sum_n` that for any given value, say n, computes the sum of the integers from 1 to n (inclusive). Use the function to determine the sum of integers from 1 to 5,000.

24. After running the code below, what is the value of `x`?

```
x <- 3
my_func <- function(y){
  x <- 5
  y+5
}
```

25. Write a function `compute_s_n` that for any given n computes the sum $S_n = 1^2 + 2^2 + 3^2 + \ldots n^2$. Report the value of the sum when $n = 10$.